

# OHJ-1110 Laaja ohjelmointi 1, tentti

20.12.2012, Essi Isohanni

Kirjoita jokaiseen vastauspaperiisi selkeästi ylläolevat otsikkotiedot, nimesi, opiskelijanumerosi, vastauspaperin järjestysnumero ja jättämiesi vastauspapereiden kokonaismäärä. *Jätä vastauspapereihisi vähintään 3cm ulkoreunamarginaalit ja ensimmäiseen paperiin vähintään 10 cm yläreunamarginaali.* Kirjoita selkeällä käsialalla. Tentin tarkastaja ei ole hieroglyfien erikoisasiantuntija. Onnea tenttiin!

## Tehtävä 1

1. Seuraavaa funktiota kutsutaan siten, että parametreiksi annetaan kaksi saman kokoista taulukkoa ja niiden koko. Lisäksi tiedetään, että kumpikaan taulukoista ei sisällä samaa alkioita kahteen tai useampaan kertaan. Selitä yksinkertaisesti suomeksi, minkä toimenpiteen funktio suorittaa? (2p.)

```
bool toimenpide( int luvut1[], int luvut2[], const int KOKO ) {
    bool lopputulos = true;
    for( int i = 0; i < KOKO; ++i ) {
        bool tarkastus = false;
        for( int j = 0; j < KOKO; ++j ) {
            if( luvut1[ i ] == luvut2[ j ] ) {
                tarkastus = true;
            }
        }
        if( not tarkastus ) {
            lopputulos = false;
        }
    }
    return lopputulos;
}
```

2. Miten edellisessä kohdassa esitetyn funktion toimintaa voitaisiin tehostaa? (2p.)
3. Etsi toinen algoritmi, jolla sama toimenpide voitaisiin suorittaa. Selosta algoritmin idea lyhyesti suomeksi. (1p.)
4. Vertaile ensimmäisen kohdan algoritmin ja edellisessä kohdassa keksimääsi algoritmin toimintaa keskenään. (2p.)
5. Kerro lyhyesti suomeksi, mitä muutoksia ensimmäisen kohdan funktioon pitäisi tehdä, jos ei voitaisi olettaa, että taulukot eivät sisällä samaa alkioita useampaan kertaan, mutta funktion pitäisi silti suorittaa sama toimenpide. (1p.)

## Tehtävä 2

1. Mitä ovat seuraavassa ohjelmakoodissa Juttu ja j? (1p.)

```
struct Juttu { int a; int b; };
Juttu j = { 42, 42 };
```

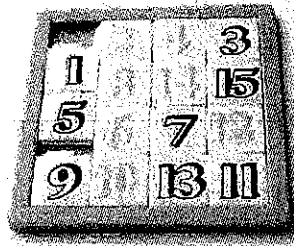
- Seuraava ohjelmapätkä on jatkoa edelliselle kohdalle. Se toimii väärin. Mitä vikaa siinä on? Miten vika ilmenee? (2p.)

```
vector< Juttu > jutut;  
jutut.at( 0 ) = j;
```

- Tarvitaan C++-ohjelma, jolla voidaan seurata Mөлky-pelin etenemistä. Jokaisella pelaajalla on numero, jonka perusteella pitää pystyä selvittämään hänen nimensä ja jokaisella kierroksella suorittamastaan heitosta saamansa pistemäärä. Määrittele C++-rakenteet, joita käyttäisit tietojen tallentamiseen. (3p.)
- Kirjoita C++:lla funktio, joka tulostaa edellisessä kohdassa määritellystä tietorakenteesta kaikkien niiden pelaajien nimet, jotka ovat jollakin pelikierroksella ylittäneet yli 10 pisteen arvoiseen heittoon. (3p.)
- Kirjoita C++:lla funktio, joka lisää kohdassa 2. määrittelemääsi tietorakenteeseen tietyn numeroiselle pelaajalle uuden heittotuloksen. (3p.)

### Tehtävä 3

Vanhanaikaisessa numeropelissä (ks. kuva) muovisen numerolaatan voi liu'uttaa viereiselle paikalle, jos siinä on tyhjä kohta. Pelin tavoitteena on saada numerolaatat suuruusjärjestykseen siten, että 1 on vasemmassa yläkulmassa ja tyhjä paikka oikeassa alakulmassa.



- Haluat toteuttaa numeropelin C++-ohjelmana, minkälaisia C++-rakenteita käyttäisit pelilaudan tallentamiseen? Perustele, miksi valitsemasi rakenne sopii parhaiten tähän tarkoitukseen. (2p.)
- Peliin tehdään yksinkertainen komentorivikäyttöliittymä, joka tulostaa pelilaudan ASCII-merkeistä muodostuvana kuviona ja kysyy käyttäjältä, minkä numeron tämä haluaa seuraavaksi siirtää tyhjälle paikalle. Suunnittele ohjelmaan funktiojako. Kerro jokaisesta funktiosta sen kuvaava nimi ja toiminta lyhyesti lauseella (tai parilla). (4p.)
- Ohjelman testaamisessa keskitytään ensimmäiseksi laatan siirtämisen oikeaan toimintaan. Suunnittele, miten testaus pitäisi suorittaa. Kerro jokaisesta testistä, mitä se testaa. (4p.)